EDA PokeData Report:

In this project I performed Exploratory Data Analysis (EDA) on Pokemon data collected from the PokeAPI, which is an open-source service containing a plethora of detailed information about Pokemon species, types, stats and more. I was initially drawn to this API due to how comprehensive it is, meaning the quality, amount and overall value of the data I can gather is excellent. The goal of my analysis was to look at this data scientifically, using the Pokemon data to come across interesting insights and anomalies when comparing them to real math and physics. This led me to explore and calculate various attributes of Pokemon, including their BMI and kinetic energy.

PokeData application:

The domain of operation of the data I have gathered could include scientific research on the physical properties of Pokemon. With a focus on translating Pokemons into something scalable with the use of their in-game stats and measurements. Likewise, it could be used by Pokemon fans to track how Pokemon stats and physical properties have changed through generations of Pokemon games, providing various fields for each Pokemon such as their stats, weight, shape and colour. It might also have a use case as data to create scale replicas of Pokemon.

Interesting Results:

Result 1: correlation between Base Stats and Kinetic Energy:

One of the first questions you ask with the data I have gathered is if the Pokemon in-game stats actually have any real effect on their physical attributes. Do Pokemon with greater physical strength or vitality actually have more theoretical energy? The scatter plot showed that there is a correlation, with the analysis revealing Pokemon with higher stats tend to generally have greater kinetic energy.

Result 2: The average kinetic energy by Pokemon type:

The bar chart was interesting, as you could derive from it that Pokemon types with the highest energy are likely to be the fastest and biggest comparatively, which makes complete sense. This, along with our knowledge from the first result, allows us to make safe assumptions about which type might be the strongest. With dragon types dominating over other types, it likely that there are many more powerful dragon type Pokemon compared to others. This might also highlight how different Pokemon types might have different battle strategies and roles due to their varying physical attributes, such as a slow Pokemon with a large health pool being used as a shield.

Result 3: Changes in base stats across generations:

This line graph revealed an interesting trend, with Pokemon from newer generations usually having higher base stats on average. This indicates that as the Pokemon franchise evolved, it tended to buff up their new additions. This might reflect advancements in gameplay for the newest generations, with stronger stats being influenced by new mechanics and design. On the other hand, this could also be a corporate trick to lean Pokemon fans towards their new content through slightly inflated, more powerful Pokemon. Other than that, this may generally indicate a shift to more competitive and higher performance gameplay in the Pokemon franchise.

Connection to lecture material:

I closely followed lecture material when developing the EDA. Tightly following methods discussed such as html requests for API calling, or use of data storing techniques with csv and the use of pandas. Checks for preparing data thoroughly like discussed within lecture – removing unnecessary text with cleaning techniques and mapping e.g. converting Pokemon generations to a more easily manageable format. Removing null values with df.dropna and transforming data into new data, adding my own derived columns. Plotting data onto graphs with the use of matpot.

Advanced Techniques Used

API Integration: The integration of an external API, gathering real-time, large-scale data about Pokémon by sending HTTP requests for a list of Pokémon, and subsequently, detailed information about each one, grabbed by looping though said data and filtering out only the essential data.

Derived Features and Calculations: Derived features, such as Body Mass Index (BMI), base stats, and kinetic energy, were calculated based on the raw data. Converted to more manageable units (kilograms and meters) before calculating BMI and kinetic energy. This enhanced the analysis, transforming the data into more meaningful metrics by adding more depth and providing insights beyond what was available in the raw dataset.

Data cleaning and transformation: The use of pandas functions like. groupby(), .dropna(), and .explode() was used to clean, organize and reshape the data for analysis. Missing values were removed from essential fields like abilities, moves, and types. I also mapped Pokémon generations from string values to numerical values, making it easier to analyse trends across generations, and used the explode() function to split multiple types into individual rows, making it much easier to analyse the impact of Pokémon types on various attributes like kinetic energy.

Data Analysis and Visualization: Various techniques to analyse the data and uncover interesting patterns. Plotting information into multiple different formats, allowing for better visualization and analysis. Using cleaned data and the matplot library, creating a scatter plot, bar chart and line chart respectively.

Reference: PokeAPI: https://pokeapi.co/